IN4394
WIRELESS NETWORKING
Dynamic Rate Control Algorithm

April 4, 2018

Himanshu Shah (4739779)
Priyanka Bhat (4735153)

Introduction

The goal of the assignment is to design a closed loop adaptive rate control algorithm for IEEE 802.11ac that selects the desirable coding scheme and rate based on feedback from the receiver. The algorithm made would be assessed using metrics: Average throughput and packet error rate.

For our algorithm, we used the estimated SNR and BER as parameters that would determine the coding scheme. We considered two algorithms, one performs good when there are large changes between SNR(larger than or equal to 5), and second when the SNR changes are small between consecutive packets. So, we used a combination of both the algorithms i.e when fluctuations are large we use one algorithm and other when the fluctations in SNR are small. To select which algorithm should be used, we observe five previous packets to determine the nature of the channel and chose an algorithm accordingly. Each algorithm is chosen for five packets and then channel behavior is analyzed again

Working

Algorithm 1 : For Large changes in SNR

When it is observed that the SNR between subsequent packets is changing by a large value (typically >= 5) then this algorithm is preferred. If the estimated SNR is greater than the upper SNR threshold value related to the current MCS, then the MCS value must be increased. First, the new desired MCS value (hence, the new snrIndex) is obtained, and the difference between the current index value and the new index value is calculated. If the difference is big (i.e, greater than 3), we increment the current snr index by 3, else increment by 1. Since, the channel variability is unpredicatable, we make sure that we don't increase the MCS value too close to the desired value, instead keeping it at a lower MCS value could protect it from bit error.

If the estimated SNR is lower than the lower SNR threshold value related to the current MCS, then the MCS value must be decreased. Here, we decrease the snr index directly to the desired index value as it would be less risky in terms of BER than decreasing it by 3 or by 1. The threshold ranges for consecutive MCS value are seen to overlap in most of the cases. The logic also tries to look at the corner cases of overlapping thresholds. If an SNR is in this overlap zone, it could be given the current MCS value or the previous MCS value. Since, we are considering the scenario where channel has large variations in SNR, the logic gives the packet the one lower MCS value. This addition to the algorithm helps decrease the number of packets having a bit error.

Algorithm 2 : For Small changes in SNR

In this algorithm, estimated SNR of the current packet is determined then the algorithm checks the MCS index to which the SNR belongs. After determining the MCS index, a switch is made to that MCS value directly. In case of SNR appearing in the overlapping threshold, a lower MCS value is preferred.This algorithm tries to select the best MCS value to get the best throughput.

Switching between Algorithms

The optimal algorithm for the channel is chosen by taking into account the BER of the previous 5 packets. Next, it determines if the majority of packets, i.e 3 or more, within the 5 packet window have bit error rate greater than zero, then the channel is considered to be having large fluctuations and algorithm 1 is chosen for the next five packets. On the other hand, if majority of packets have bit error rate as zero, then the channel is considered to have small SNR changes and algorithm 2 is chosen.

Deciding Parameters

As we are using a combination of two algorithms between which we would be switching continuously based on the channel behavior, we needed to decide at what frequency we would be switching and which algorithm to start with. To determine the number of packets that should be analyzed before switching the algorithm, we experimented with different values and observed their result. Based on the observation, we saw that packet window of 3 and 5 were good but 5 performed better than three on many occasions. So, we selected a window of 5 packets to determine whether to switch between algorithms after every interval of five packets.

To decide which algorithm to start with, we experimented by running each algorithm first. Based on the results, we saw that on starting with algorithm 2 throughput for one of the value was 25.45Mbps, whereas on starting with algorithm 1 throughput was 24.27Mbps, but Packet Error Rate remained constant. As performance was better for most cases when algorithm 2 was used as starting algorithm, we decided to start with algorithm 2. Another observation that we made that even though MCS9 is not supported when channel bandwidth is 20MHz, original code tries to push MCS value to 9 which results in an error. We have taken this into account in our algorithm and prevent this error by placing a check for the condition and not letting MCS value go beyond 8 in case of 20MHz bandwidth.

Observation

To test the performance of both the algorithms we provided a range of inputs such as all the combinations of channel and bandwidth, changing number of packets from 10 to 300, providing a manual random SNR data array, varying maxJump from 0.5 to 5. Plots for some of the data are shown below, and a link to the complete test dataset and its outcome are provided in appendix. For majority of the cases our algorithm showed a better throughput with reduction in Packet Error Rate simultaneously. Even for the cases which had lower throughput, bit error rate for the proposed algorithm was found to be on the lower side. Hence, there was a gain in at least one of the parameters in majority of the scenarios.

Results

We observed that our proposed algorithm performed better than the original algorithm on 75% of the cases and for 16% of the cases original algorithm performed better. There were 9% cases when the performance of the two algorithms were same or comparable. Figure

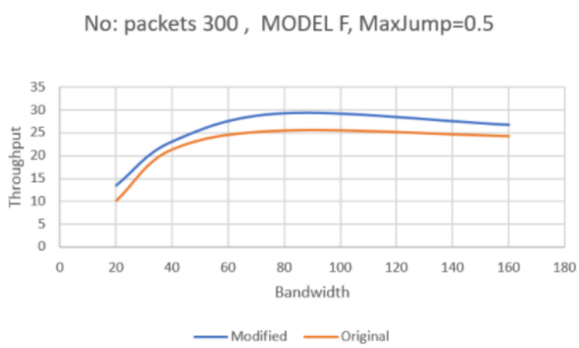

Figure 1: Plot for Throughput vs. Bandwidth

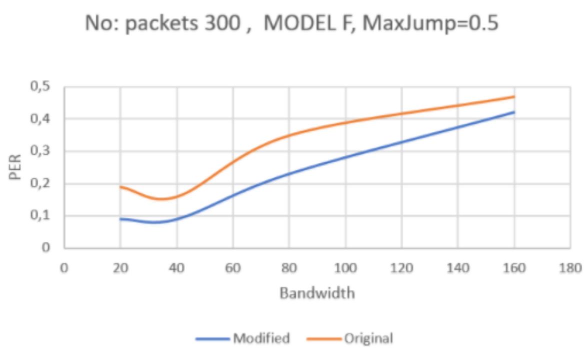

Figure 2: Plot for PER vs. Bandwidth

1,2 and Figure 3,4 show the comparison in performance of both algorithms for various channel bandwidth for MODEL-F delay profile transmitting 300 packets with the MaxJump parameter=0.5 and MaxJump parameter=5. We can notice from the graph that there is gain in throughput whereas at the same time packet error rate is reduced.

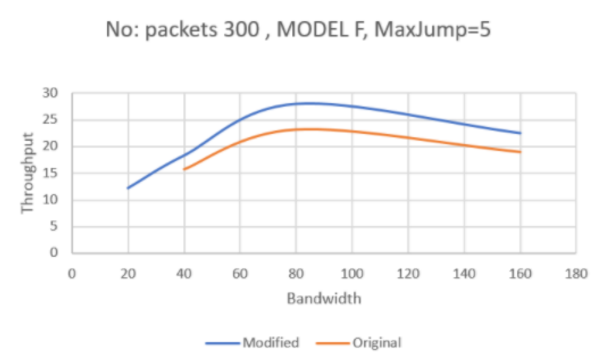

Figure 3: Plot for Throughput vs. Bandwidth

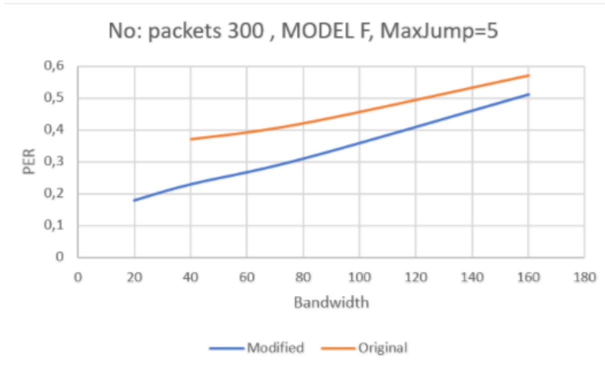

Figure 4: Plot for PER vs. Bandwidth

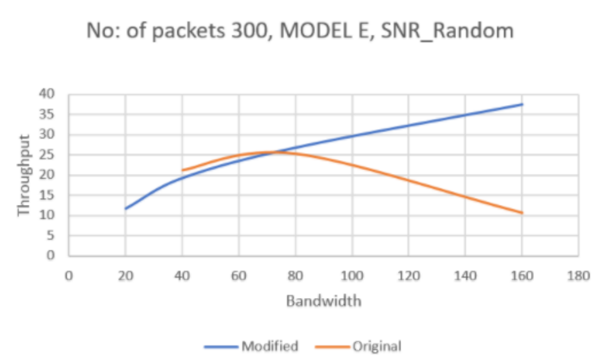

Figure 5: Plot for Throughput vs. Bandwidth

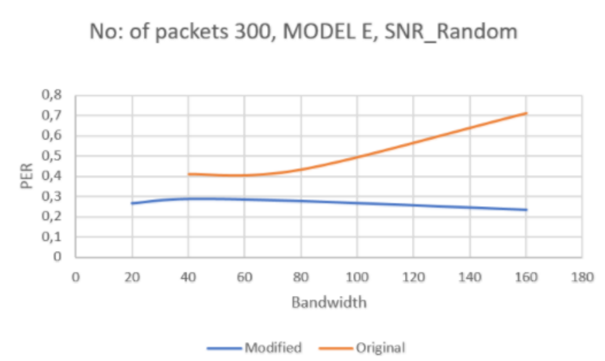

Figure 6: Plot for PER vs. Bandwidth

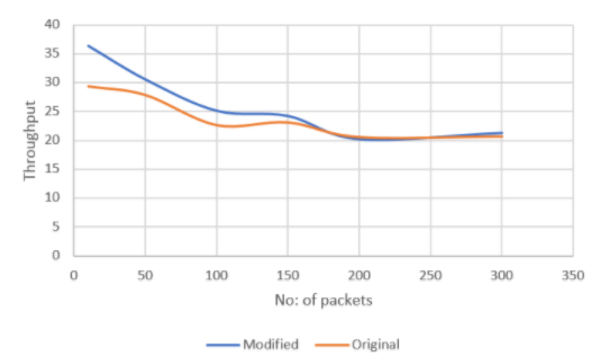

Figure 7: Plot for Throughput vs. Bandwidth

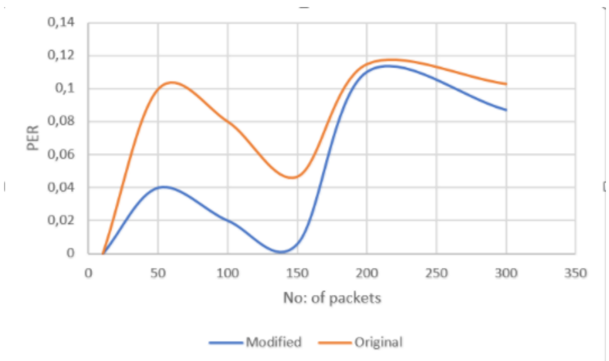

Figure 8: Plot for PER vs. Bandwidth

Figure 5,6 show a similar trend when tested against hard coded random SNR values that varied extremely. In figure 5, we can notice that at two instances the throughput of the proposed algorithms is lower than the original algorithms, but we can see that there is a large difference in packet error rate which makes up for the lower throughput. Figure 7,8 shows the performance for the two algorithms when number of packets are varied. In this case also we can see higher or similar throughput but with lower Packet error rate.

Conclusion

In this assignment we designed a closed loop adaptive rate control algorithm for IEEE 802.11ac and tested it against the provided algorithm. The performance of both the algorithms were studied and compared and the proposed algorithm emerges to be better on majority of the scenarios.

Appendix

Link for the test dataset and results:

https://bit.ly/2ItK2sw